МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА − Российский технологический университет»

РТУ МИРЭА

Институт информационных технологий (ИИТ)
Кафедра цифровой трансформации (ЦТ)

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ

по дисциплине «Разработка баз данных»

Практическое занятие № 5

Студенты группы *ИКБО-42-23 Голев С.С.*
.

_____
(подпись)

Ассистент *Морозов Д.В.*

_____
(подпись)

Отчет представлен «___»_____2025 г.

Москва 2025 г.

СОДЕРЖАНИЕ

ЗАДАНИЕ

Задание №1: создание модифицируемого представления Для вашей базы данных создать простое модифицируемое представление, которое отбирает строки из одной таблицы по определенному критерию.

Например, для БД «Аптека» можно создать представление, отображающее лекарства только от одного производителя «Bayer AG»).

Задание №2: модификация данных через представление

Продемонстрировать возможность изменения данных в базовой таблице через представление, созданное в Задании №1. Для этого необходимо выполнить два запроса:

1. Добавить новую запись с помощью оператора INSERT.

2. Удалить существующую запись с помощью оператора DELETE.

Задание №3: создание немодифицируемого аналитического представления Для вашей базы данных создать единое немодифицируемое представление для аналитических целей.

Представление должно объединять данные как минимум из двух таблиц и содержать агрегирующие функции (COUNT, SUM, AVG и т.д.) и группировку (GROUP BY).

Например, для БД «Аптека» такое представление могло бы для каждого производителя выводить общее количество наименований лекарств и их среднюю цену.

Задание №4: использование аналитического представления в запросах

Написать SELECT-запрос, который использует созданное в Задании №3 аналитическое представление в качестве источника данных для дальнейшей фильтрации или анализа.

Например, можно отобрать производителей, у которых средняя цена на продукцию превышает определенное значение.

Задание №5: Создание и обновление материализованного представления

1. Создать материализованное представление для ускорения выполнения ресурсоемкого аналитического запроса.

Например, для БД «Аптека» можно создать представление, которое заранее рассчитывает общую сумму продаж для каждого покупателя.

2. Продемонстрировать процесс обновления данных в представлении с помощью команды REFRESH MATERIALIZED VIEW viewName;.

Задание №6: разработка пользовательской функции для аналитических вычислений

1. Разработать пользовательскую функцию, которая инкапсулирует комплексный аналитический расчет. Функция должна принимать на вход идентификатор (например, manufacturer_id) и возвращать одно скалярное значение (например, общую сумму продаж продукции данного производителя), вычисленное на основе соединения нескольких таблиц и применения агрегатных функций.

2. Продемонстрировать вызов функции в составе SELECT-запроса.

Задание №7: разработка хранимой процедуры для выполнения сложной операции Разработайте хранимую процедуру, которая выполняет безопасную операцию по изменению данных. Процедура должна принимать на вход ID какой-либо записи и числовое значение (например, количество). Внутри процедуры необходимо проверить, достаточно ли текущего значения в числовом поле одной таблицы для выполнения операции.

• Если да – уменьшите это значение и добавьте новую запись в другую, связанную таблицу.

• Если нет – операция должна полностью прерваться, не внося никаких изменений в данные.

Для сообщения о результате используйте выходной параметр, который вернёт статус успеха или неудачи.

Задание №8: демонстрация вызова хранимой процедуры

Привести два примера вызова процедуры, созданной в Задании №7:

• Успешный вызов, который добавляет в вашу базу данных уникальную

запись.

• Неудачный вызов, который демонстрирует срабатывание реализованной проверки целостности и возврат пользовательской ошибки.

Каждый SQL-запрос сопроводить комментарием, объясняющим его назначение и логику работы с учетом специфики вашей базы данных.

ВЫПОЛНЕНИЕ ЗАДАНИЯ

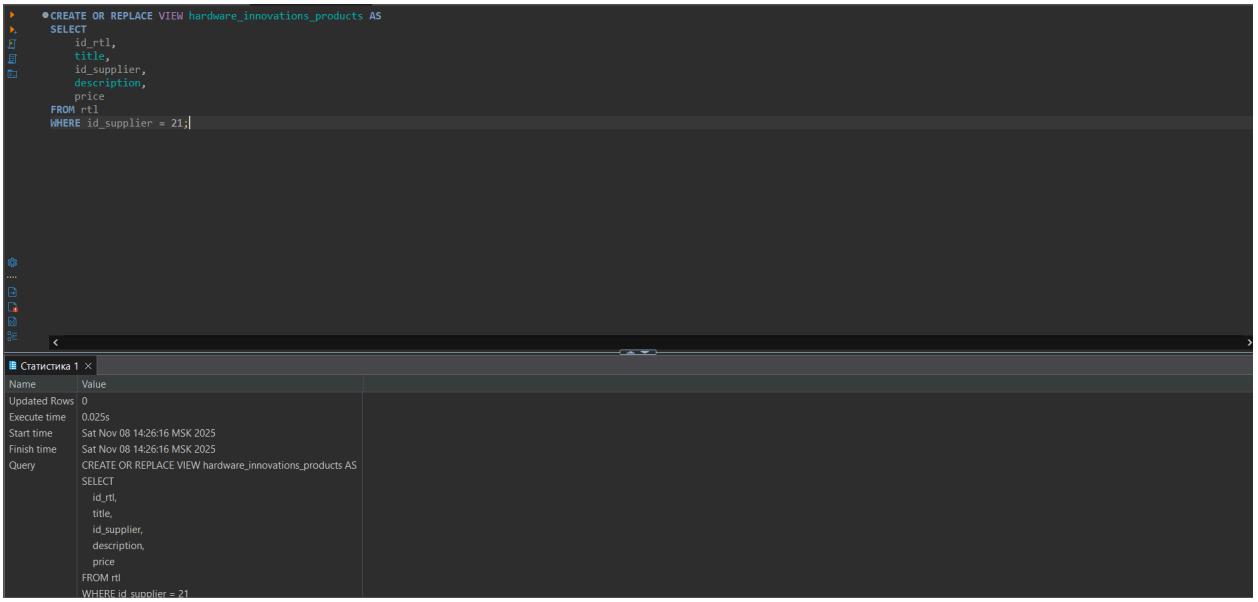


Рисунок 1 – создание модифицированного представления

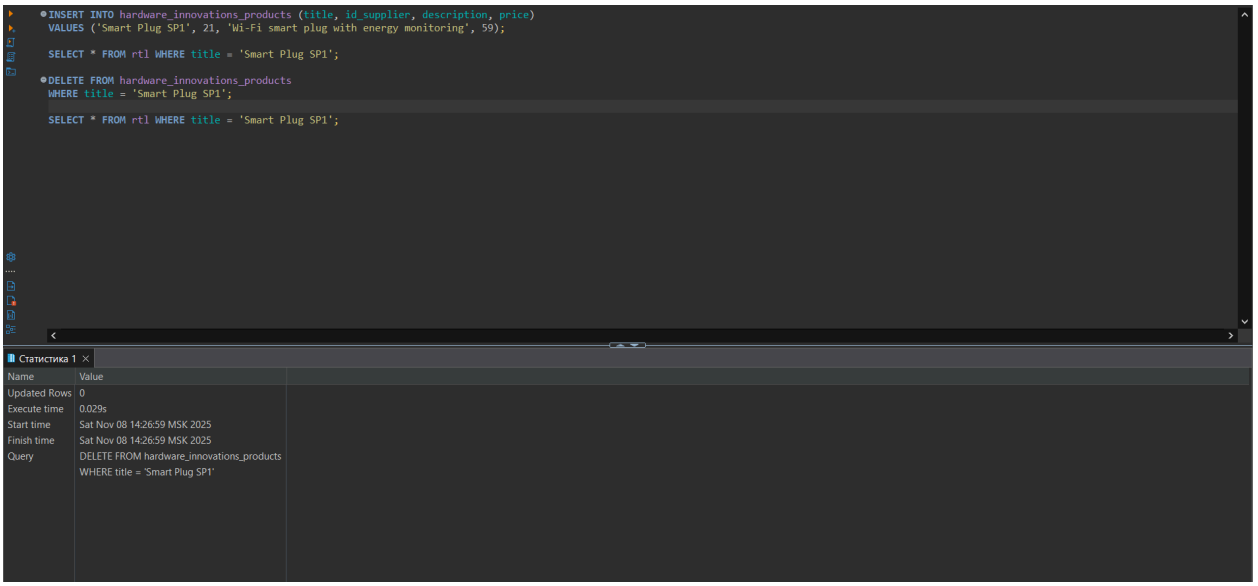


Рисунок 2 – модификация данных представления

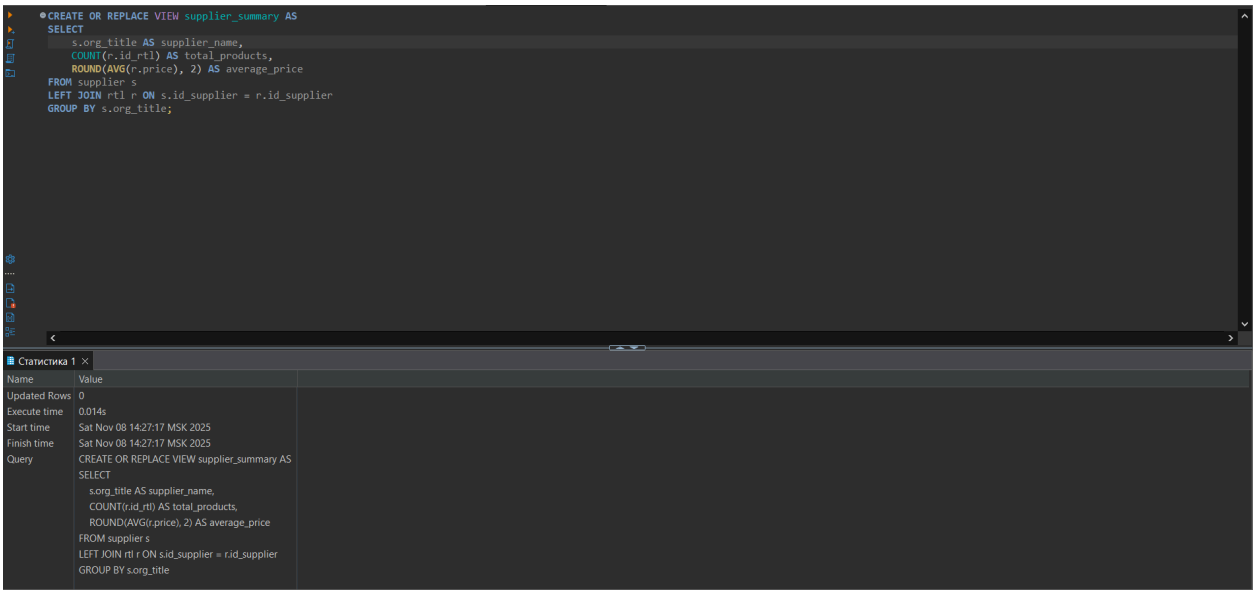


Рисунок 3 – создание немодифицированного аналитического представления

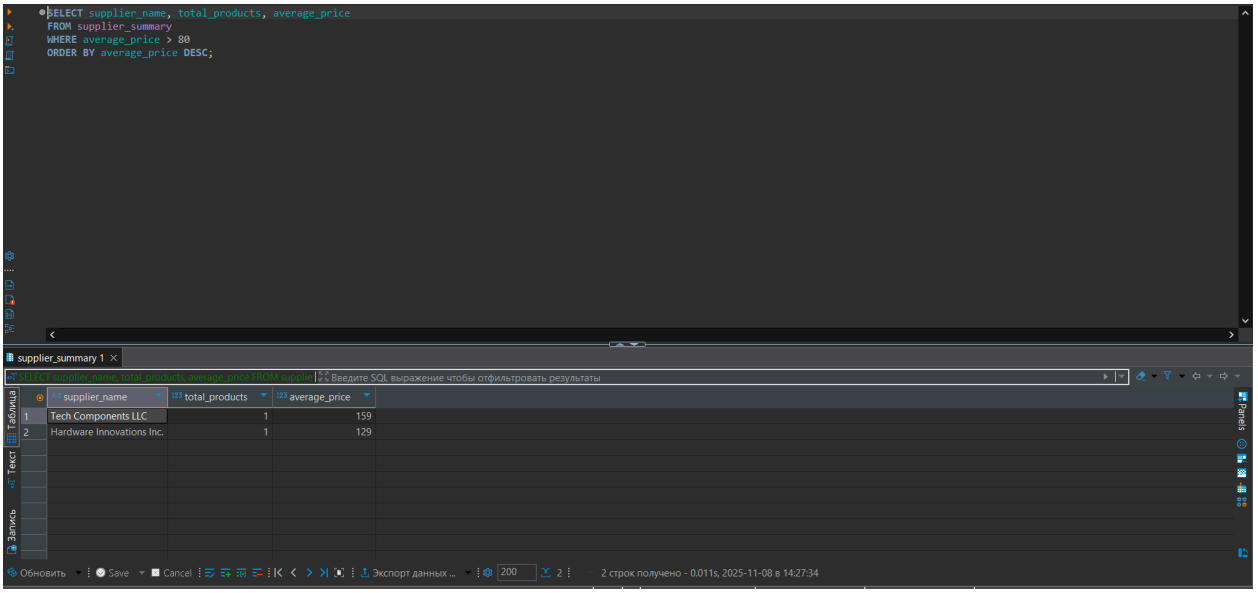


Рисунок 4 – использование аналитического представления в вопросах

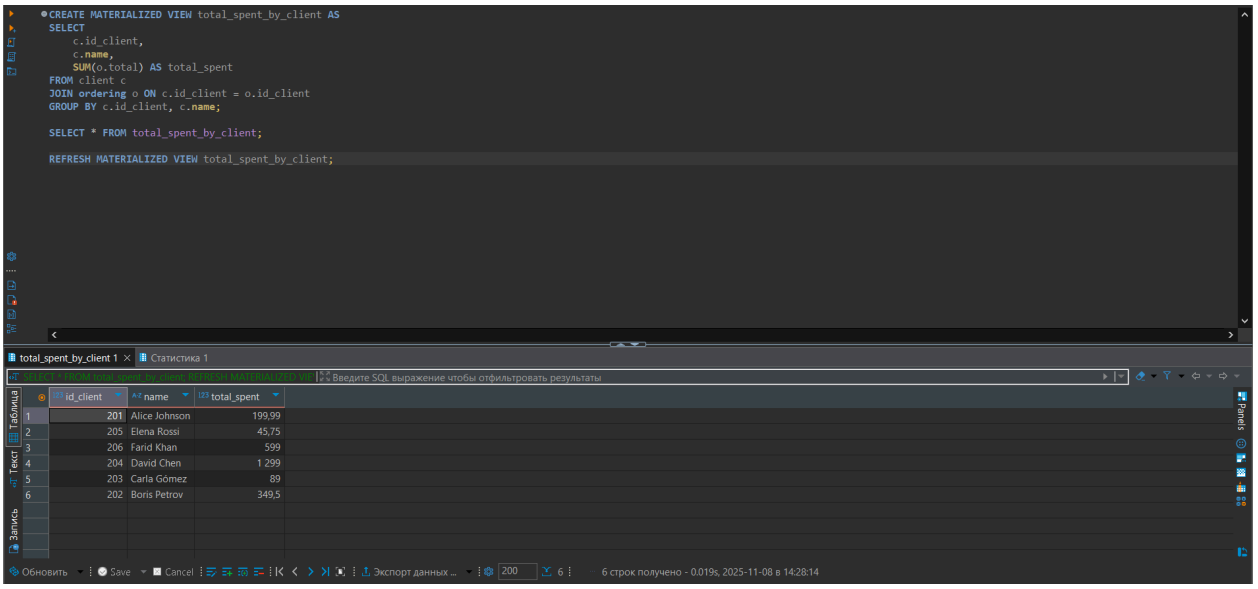


Рисунок 5 – Создание и обновление материализованного представления

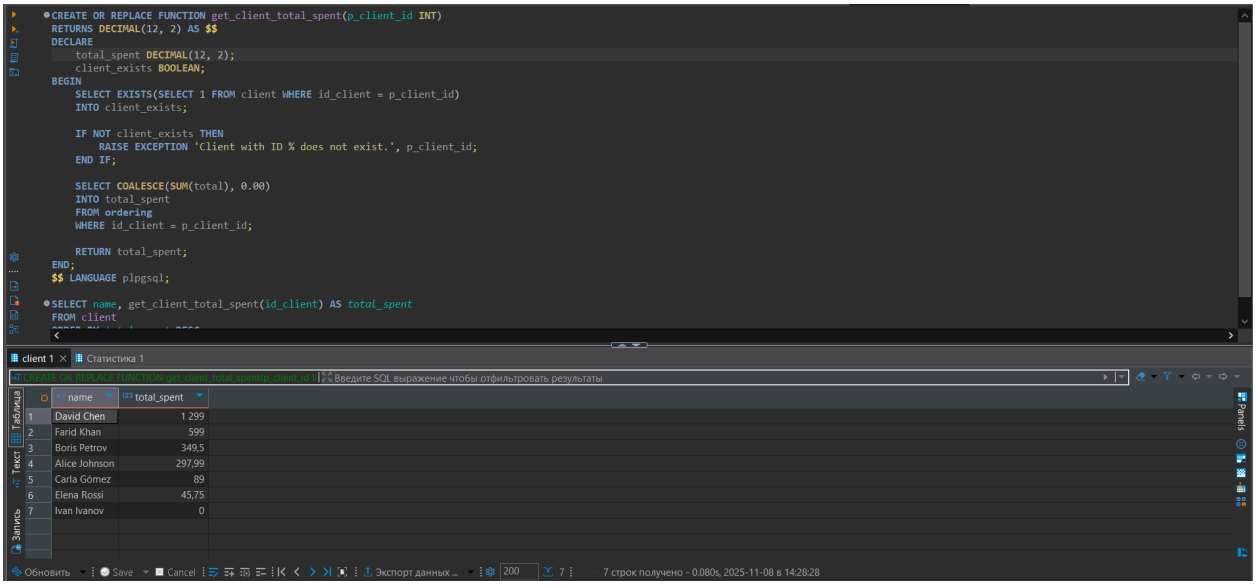
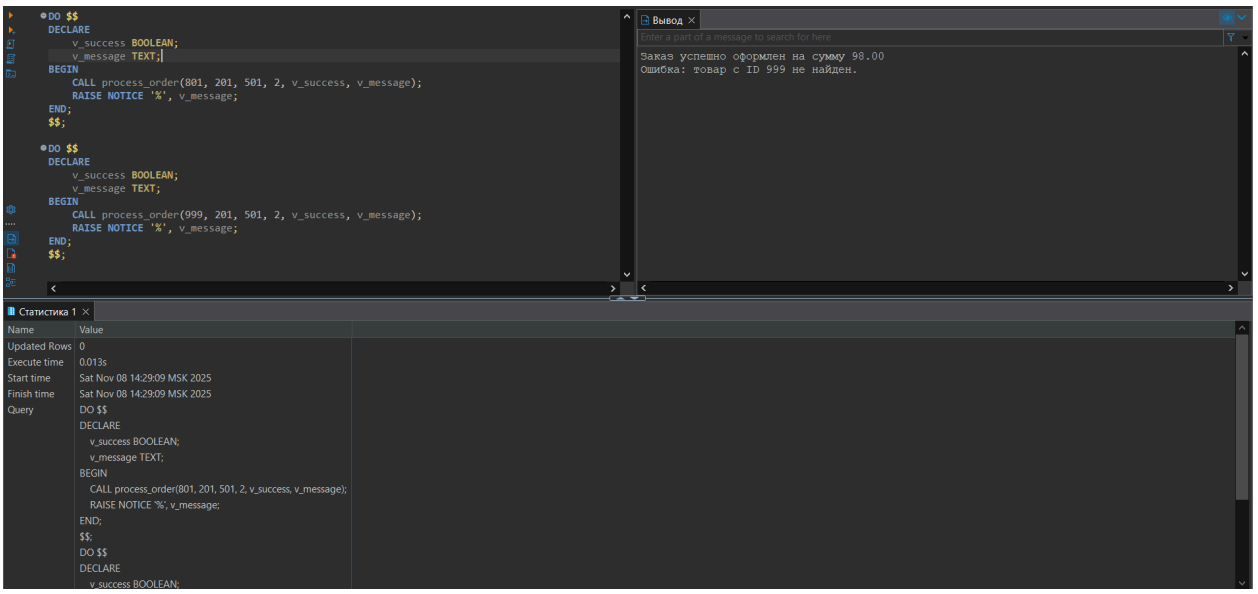


Рисунок 6 – Разработка пользовательской функции для аналитических вычислений

```
CREATE OR REPLACE PROCEDURE process_order(
    p_rtl_id INT,
    p_client_id INT,
    p_employee_id INT,
    p_quantity INT,
    OUT p_success BOOLEAN,
    OUT p_message TEXT
)
LANGUAGE plpgsql AS $$
DECLARE
    current_price DECIMAL(10,2);
    total_sum DECIMAL(10,2);
    product_exists BOOLEAN;
BEGIN
    SELECT EXISTS(SELECT 1 FROM rtl WHERE id_rtl = p_rtl_id)
    INTO product_exists;

    IF NOT product_exists THEN
        p_success := FALSE;
        p_message := 'Ошибка: товар с ID ' || p_rtl_id || ' не найден.';
        RETURN;
    END IF;

    SELECT price INTO current_price FROM rtl WHERE id_rtl = p_rtl_id;

    total_sum := current_price * p_quantity;

    INSERT INTO ordering (id, id_ordering, id_employee, id_address, id_client, total, date, status, delivery_method)
    VALUES (
        (SELECT MAX(id)+1 FROM ordering),
        (SELECT MAX(id_ordering)+1 FROM ordering),
        p_employee_id,
        (SELECT id_address FROM client_address WHERE id_client = p_client_id LIMIT 1),
        p_client_id,
        total_sum,
        CURRENT_DATE,
        'paid',
        'courier'
    );

    p_success := TRUE;
    p_message := 'Заказ успешно оформлен на сумму ' || total_sum;
END;
```

Рисунок 7 – Разработка хранимой процедуры для выполнения сложной операции



```
DO $$
DECLARE
    v_success BOOLEAN;
    v_message TEXT;
BEGIN
    CALL process_order(801, 201, 501, 2, v_success, v_message);
    RAISE NOTICE '%', v_message;
END;
$$;

DO $$
DECLARE
    v_success BOOLEAN;
    v_message TEXT;
BEGIN
    CALL process_order(999, 201, 501, 2, v_success, v_message);
    RAISE NOTICE '%', v_message;
END;
$$;
```

Вывод

```
Заказ успешно оформлен на сумму 98.00
Ошибка: товар с ID 999 не найден.
```

Статистика 1

Name	Value
Updated Rows	0
Execute time	0.013s
Start time	Sat Nov 08 14:29:09 MSK 2025
Finish time	Sat Nov 08 14:29:09 MSK 2025

Query	DO $$
	DECLARE
	v_success BOOLEAN;
	v_message TEXT;
	BEGIN
	CALL process_order(801, 201, 501, 2, v_success, v_message);
	RAISE NOTICE '%', v_message;
	END;
	$$;
	DO $$
	DECLARE
	v_success BOOLEAN;

Рисунок 8 – Демонстрация вызова хранимой процедуры